

# Consistency Checking Rules of Variability between Feature Model and Elements in Software Product Lines

Se-Hoon Kim<sup>†</sup> · Jeong-Ah Kim<sup>††</sup>

## ABSTRACT

Many companies have tried to adopt Software Product Line Engineering for improving the quality and productivity of information systems and software product. There are several models defined in software product line methodology and each model has different abstraction level. Therefore it is important to maintain the traceability and consistency between models. In this paper, consistency checking rules are suggested by traceability matrix of work products.

**Keywords :** Software Product Line, Feature, Feature Model, Variability, Consistency

## 소프트웨어 제품라인의 휘처모델과 구성요소간 가변성에 대한 일관성 검증 규칙

김 세 훈<sup>†</sup> · 김 정 아<sup>††</sup>

## 요 약

모든 기업들은 높은 품질의 정보시스템과 높은 생산성을 가지는 소프트웨어 제품을 만들기 위해 소프트웨어 제품라인 공학(software product line engineering)을 도입하고 있다. 소프트웨어 제품라인 방법론은 다양한 모델들을 가지고 있으며, 각 모델은 추상화 관점과 수준이 서로 다르다. 이러한 모델에 존재하는 요소들간 추적성(traceability)과 가변성(variability) 정보의 일관성(consistency)을 유지하는 것이 중요하다. 본 연구에서는 휘처(feature)의 가변성과 다른 산출물에 정의한 가변성의 일관성을 검증하는 규칙을 제시하였다.

**키워드 :** 무선랜, 소프트웨어 제품라인, 휘처, 휘처모델, 가변성, 일관성

## 1. 서 론

모든 기업들은 높은 품질의 정보시스템과 높은 생산성을 가지는 소프트웨어 제품을 만들기 위해 노력하고 있다. 이를 달성하기 위하여 소프트웨어 제품라인 공학을 도입하려고 시도하는 기업들이 증가하고 있다. 소프트웨어 제품라인 공학기반 개발은 시장적시성과 개발비용을 현저하게 감소시킬 수 있는 방법이라고 평가받고 있기 때문이다[1, 2, 3].

소프트웨어 제품라인 공학은 하나의 애플리케이션이 아닌 하나의 도메인(domain)에 속한 다양한 유사 제품들을 대상으로 한다. 휘처의 선택과 제품별 가변성을 수용하는 커스터마이제이션(customization) 과정을 통해 고객 니즈(needs)

별 제품을 생산할 수 있는 방법론이다. 제품간 공통성(commonality)과 가변성을 휘처로 분석하고 재사용 가능한 핵심 자산(core asset)과 아키텍처를 구축하는 도메인 공학(domain engineering)과 만들어진 자산들을 조립하여 새로운 시스템을 만드는 애플리케이션 공학(application engineering)으로 이루어진다. 도메인 공학 프로세스의 각 단계별 산출물을 구성하는 구성 요소는 가변성 정보를 속성으로 가지고 있고 이러한 산출물 간에는 추적성을 수립해야 한다.

IEEE(Institute of Electrical and Electronics Engineers)에서는, 개발 과정에서 만들어진 서로 다른 둘 이상의 산출물간 연결성을 정의하는 것으로 자동화, 일관성, 응집성을 강화하기 위한 수단을 추적성이라고 정의하고 있다[6]. 일반적인 소프트웨어 공학에서의 추적성은 사용자의 요구사항이 구현물에 반영이 되어있는지 확인할 수 있는 것을 의미한다.

소프트웨어 제품 커스터마이제이션은 선택된 제품 휘처들을 실현하는 핵심 자산들과 추가모듈을 선택하고 조립하는 활

<sup>†</sup> 준 회 원 : 에듀크레프트 대표이사  
<sup>††</sup> 종신회원 : 관동대학교 컴퓨터 교육과 교수  
논문접수 : 2013년 10월 22일  
수정일 : 1차 2013년 11월 22일  
심사완료 : 2013년 11월 23일  
\* Corresponding Author : Jeong-Ah Kim(clara@kd.ac.kr)

동이다. 그러므로 제품라인의 모든 휘처와 핵심자산 및 추가모듈간의 추적성은 중요하다. 또한, 제품라인의 휘처는 가변성 정보를 포함하고 있기 때문에 휘처와 추적관계를 갖는 핵심자산 및 선택적 모듈의 가변성도 일관성을 유지해야 한다[4].

본 연구에서는 소프트웨어 제품 커스터마이제이션의 정확성 증대를 위해 단계간 추적성 수립 방법을 정의하였다. 또한, 휘처와 휘처를 구현한 요소간 가변성의 일관성 유지 원칙을 정의하고, 모델링 과정에 적용할 수 있는 검증규칙으로 정형화 하였다.

2장에서는 가변성 모델링 방법과 기존의 추적성 검증기법에 대해 알아보고, 3장에서는 추적성 및 일관성을 검증할 수 있는 규칙을 제안한다. 4장에서는 적용한 사례를, 마지막 5장에서는 결론을 제시한다.

## 2. 관련 연구

### 2.1 가변성 모델링

소프트웨어 제품라인 공학은 같은 도메인에 속한 소프트웨어 제품들간 공유할 수 있는 핵심 자산을 구축하고, 구축한 핵심 자산을 기반으로 소프트웨어를 개발하는 방법이다[3]. 이를 위해서는 다양한 제품 간 공통성과 가변성을 파악하고 관리하는 것이 제품라인 공학에서의 핵심이라 할 수 있다[1, 5].

공통성이란 같은 도메인에 속한 모든 소프트웨어 제품들이 가지는 공통적인 특징이고, 가변성이란 같은 도메인에 속한 제품들 중에서 일부 제품들이 가지는 특징이다. 공통성과 가변성은 휘처를 사용하여 표현을 한다.

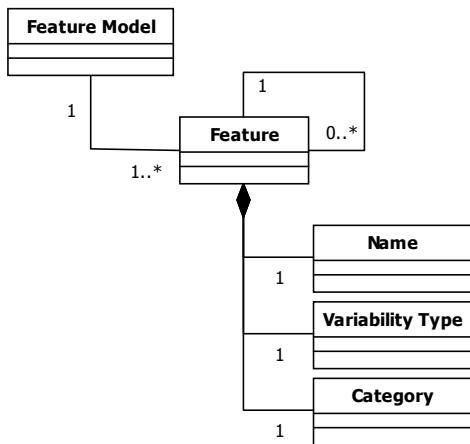


Fig. 1. Element of Feature Model

휘처는 도메인에 속하는 제품들간 공통점과 차이점을 정의하는 단위로, 사용자나 개발자가 식별할 수 있는 제품의 특징을 의미[4, 5]하며 구성요소는 Fig. 1과 같다. 휘처의 속성으로는 이름, 가변성 유형, 범주가 있다. 가변성 유형(variability type)으로는 모든 제품이 만족해야 하는 필수적

(mandatory) 유형, 제품이 포함하거나 포함하지 않을 수 있는 선택적(optional) 유형, 선택 가능한 목록 중에서 하나를 포함할 수 있는 대안적(alternative) 유형이 있다. 휘처는 4가지 범주(category)로 구분하는데 역량(capability), 운영환경(operating environment), 도메인 기법(domain technology), 구현기술(implementation technique)들 이다.

소프트웨어 제품라인에는 휘처를 구현한 자산과 이들을 구성하는데 기반이 되는 아키텍처가 존재한다. 아키텍처를 기반으로 이들 자산들을 재구성하여 새로운 소프트웨어 제품을 생성한다. 소프트웨어 제품 생성시 포함되어야 할 자산들은 휘처의 가변성 타입과 제품이 만족할 휘처에 의해 달라진다. Fig. 2는 휘처모델에서 원하는 휘처를 선택하면 관련 자산들을 조합하여 새로운 소프트웨어 제품을 생성하는 과정을 보여준다.

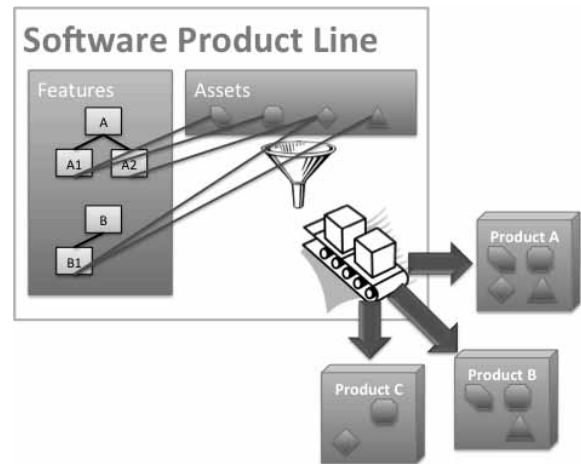


Fig. 2. Software Product line

### 2.2 정형적 일관성 검증법

소프트웨어 제품라인 개발에 있어서, 산출물간의 복잡한 관계성이 증가함에 따라 공통성과 가변성간의 일관성 추적이 더 중요해졌다. [6]에서는 휘처모델과 소프트웨어 아키텍처의 컴포넌트&커넥터 뷰 간 일관성을 증명하기 위해 정형적 일관성 검증법을 제안하였다. 제안된 검증법은 '휘처가 아키텍처의 컴포넌트와 매핑(mapping)이 된다.'는 정의(theorem)을 기반으로 일대일 매핑(one-to-one mapping), 일대다 매핑(one-to-many mapping), 다대일 매핑(many-to-one mapping), 다대다 매핑(many-to-many mapping)으로 구분하였다. 각 매핑은 Strong Checking과 Weak Checking으로 상세화된다. Strong Checking은 휘처와 구성요소 간에 가변성이 모두 동일한지 검사하고, Weak Checking은 휘처와 구성요소 간에 가변성이 최소한 1개는 동일한지 검사한다. 휘처의 가변성 타입별 Strong Checking과 Weak Checking은 Table 1과 같다.

Table 1. Strong and Weak checking by variability of feature

		Mandatory	Optional	Alternative
Strong	조건	if feature f is mandatory	if feature f is optional	if feature f is alternative
	결과	all components which implement feature f must be mandatory.	all components which implement feature f must be optional	all components which implement feature f must be alternative
Weak	조건	if feature f is mandatory	if feature f is optional	if feature f is alternative
	결과	there is at least one mandatory component	there is at least one optional component	there is at least one alternative component

### 3. 일관성 검증 규칙

도메인 공학의 분석단계 산출물인 휘처모델과 설계단계 산출물인 레퍼런스 아키텍처(reference architecture)간, 레퍼런스 아키텍처와 구현단계 산출물인 코드 간에는 상호 추적성이 필요하다. 휘처 구현을 위한 방법을 레퍼런스 아키텍처에 반영해야하고, 그 아키텍처를 구현한 코드를 만들어야 하기 때문이다.

도메인 공학의 각 단계별 산출물에는 가변성과 관련된 휘처를 식별한다. 산출물을 구성하는 구성요소에도 가변성을 표현해야 한다. 하나의 산출물을 다양한 요소를 포함하고 있으면서, 요소간의 계층관계를 형성하여 하나의 산출물 내에서의 개념적 종속관계를 형성한다. 예를 들어서 소프트웨어 아키텍처 구성요소를 나타낸 Fig. 3에서 보듯이 아키텍처를 구성하는 하나의 요소(element)는 다른 요소와 연관이 될 수도 있고 다른 요소를 하위요소로 포함할 수 있다. 그러므로, 정확한 소프트웨어 제품 구성을 위해서는 휘처와 휘처를 구현한 산출물의 가변성 정보의 일관성을 보장해야 한다. 또한 산출물이 갖는 개념적 종속관계를 고려하여 산출물 구성요소간에도 가변성 정보의 일관성을 보장해야 한다.

본 논문에서는 휘처와 산출물간 가변성 일치성을 검증하는 규칙(F-E Rules)과 계층구조를 갖는 구성요소간 가변성 일치성을 검증하는 규칙(E-E Rules)을 제시한다.

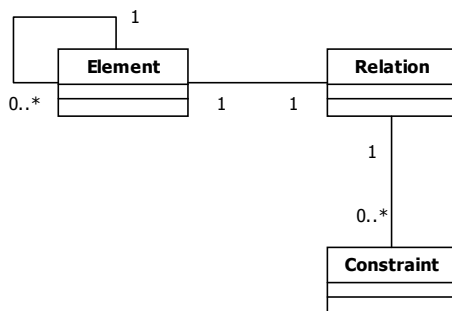


Fig. 3. Elements of Software Architecture in FORM-UML

#### 3.1. F-E Rules

F-E Rules는 휘처의 가변성 정보와 휘처를 구현한 구성요소의 가변성 정보간의 일관성을 검증하기 위한 규칙이다.

• **F-E Rule 1** : 휘처 가변성이 필수적인 경우, 가변성이 필수인 1개 이상의 구성요소가 존재

가변성이 필수적인 휘처는 모든 소프트웨어 제품이 가져야 하는 휘처이므로, 이를 구현한 구성요소도 반드시 제품에 포함시켜야 하므로 가변성 정보를 필수로 정의해야한다.

• **F-E Rule 2** : 휘처 가변성이 선택적인 경우, 휘처와 추적성을 갖는 구성요소 중 1개 이상이 가변성정보를 선택적으로 정의

가변성이 선택적인 휘처는 제품 커스터마이제이션 과정에서 제품에 포함시킬지 여부를 결정할 수 있음을 의미한다. 만약, 휘처를 선택하지 않으면 휘처와 추적성을 갖는 구현요소를 배제시킬 수 있어야 한다.

• **F-E Rule 3** : 휘처 가변성이 대안적인 경우, 휘처와 추적성을 갖는 구성요소 중 1개 이상이 가변정보를 대안적으로 정의

가변성이 대안적인 휘처는 제품에 포함시킬 대안목록 중 하나이고 그 대안목록 중에서 선택된 것만 제품에 포함할 수 있으므로, 이를 구현한 구성요소도 제품에 포함할 선택적 대안이 있어야 하기 때문이다.

#### 3.2. E-E Rules

E-E Rules는 종속관계를 갖는 요소들이 갖는 가변성간의 일관성을 검증하기 위한 규칙이다. 구성요소간 종속관계는 연결관계(association)과 같은 수평적 계층 관계와 집합관계(aggregation) 및 상속관계(inheritance)와 같은 수직적 계층관계가 있다. 구성요소들은 이들 종속관계로 인해 다른 구성요소의 가변성에 서로 영향을 받을 수 있으므로 종속관계를 갖는 요소들간의 가변성 정보도 일관성을 유지해야 한다.

• **E-E Rule 1** : 가변성이 대안적인 구성요소는 동일한 계층에 대안적 가변성을 가진 구성요소가 2개 이상 존재.

가변성이 대안적인 구성요소는 여러 대안들 중에서 하나만 선택할 수 있는 것을 의미하므로, 해당 구성요소와 동일한 계층에 대안으로 선택 가능한 구성요소가 존재해야 한다.

• **E-E Rule 2** : 가변성이 대안적인 구성요소는 필수적 가변성 또는 선택적 가변성을 갖는 상위요소의 하위요소로 존재.

가변성이 대안적인 하위요소는 상위요소의 선택에 따라 대안목록이 제품에 포함이 되지 않거나 대안목록 중에서 하나를 선택하여 제품에 포함이 될 수 있으므로, 해당 상위요소에는 대안목록의 참여여부를 판단할 수 있는 필수적이나 선택적 가변성을 갖는 상위요소가 있어야 하기 때문이다.

#### 4. 적용 사례

본 연구에서 제안한 5가지 규칙을 수업행동분석[3, 7, 8] 소프트웨어 제품라인 아키텍처에 적용하였다. 수업행동분석은 수업의 질을 개선하려는 목적으로 교사 행동요소에 대한 동료나 학생, 전문가로 부터 피드백을 받는 활동[8]이며, 각 대학에서는 취업역량을 배양하기 위해 면접행동을 분석하기 위한 모의면접장소와 모의면접시스템을 구축하는 추세이다.

Fig. 4는 수업행동분석도메인에 대한 휘처모델이다. Capability Layer에는 평가미디어, 평가하기, 평가자료보기라는 3가지 휘처가 있다. Fig. 5는 수업행동분석시스템 아키텍처와 휘처를 연결한 것이다. 이 아키텍처의 각 요소별로 스테레오타입(stereotype)을 이용하여 가변성 및 추적성을 갖는 휘처를 표기하였다. 하지만 스테레오타입에 휘처명을 모두 기입하면 아키텍처 구성요소의 크기가 비대해지므로 휘처마다 식별아이디를 적용하여 이를 기입하였다.

##### 4.1 F-E Rules 적용 예

수업행동분석 도메인 휘처모델에서 가변성이 필수적인

view Evaluation Data(평가자료보기, C2-1) 휘처는 행동분석시스템 아키텍처에서 필수적 가변성을 가진 Media Player와 선택적 가변성을 가진 Evaluation와 추적성을 갖고 있다. 그러므로 view Evaluation Data 휘처는 1개 이상의 필수적 가변성을 가지므로 휘처와 산출물간 가변성이 일치한다.

행동분석 도메인 휘처모델에서 가변성이 선택적인 do Capture(캡춰하기, C2-2-1)는 행동분석시스템 아키텍처에서 선택적 가변성을 가진 Capture와 Snapshot, 대안적 가변성을 가진 Media Player에 참여한다. 그러므로 do Capture 휘처는 1개 이상의 선택적 가변성을 가지므로 휘처와 산출물간 가변성이 일치한다.

행동분석도메인 휘처모델에서 가변성이 대안적인 Video(영상, C1-1)는 행동분석시스템 아키텍처에서 대안적 가변성을 가진 Video Player에 참여한다. 또한 휘처모델에서 대안적 가변성을 가진 Audio(음성, C1-2)는 행동분석시스템 아키텍처에서 대안적 가변성을 가진 Audio Player에 참여한다. 그러므로 Video와 Audio 휘처는 휘처와 산출물간 가변성이 일치한다.

##### 4.2 E-E Rules 적용 사례

아키텍처에서 가변성이 필수적인 Evaluation Media(평가미디어, C1)에는 Video Player와 Audio Player라는 구성요소가 포함되어 있으며 이들의 가변성은 대안적이다. 두 요소 모두 동일한 수준에 위치해있으므로 구성요소간 가변성 규칙이 일치한다.

아키텍처에서 가변성이 선택적인 Video Player와 Audio

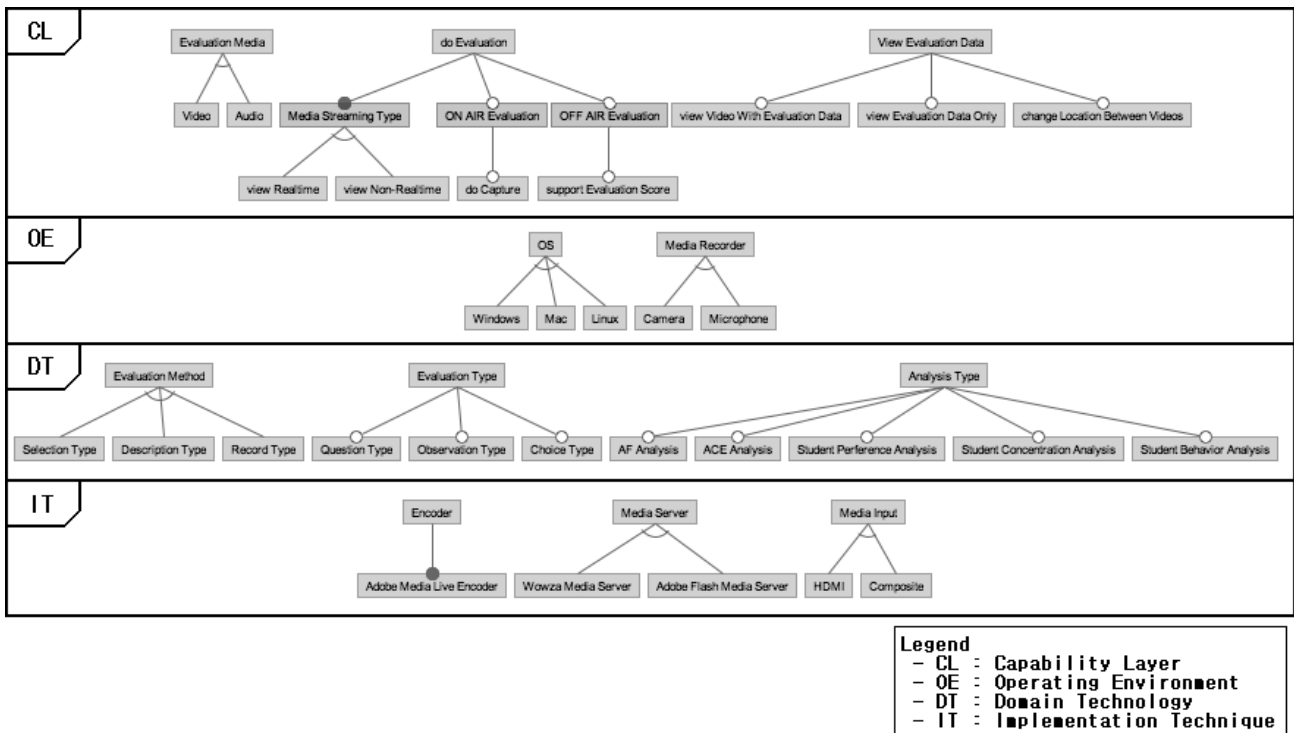


Fig. 4. Feature model of Online Evaluation Classroom Behavior Service Product line

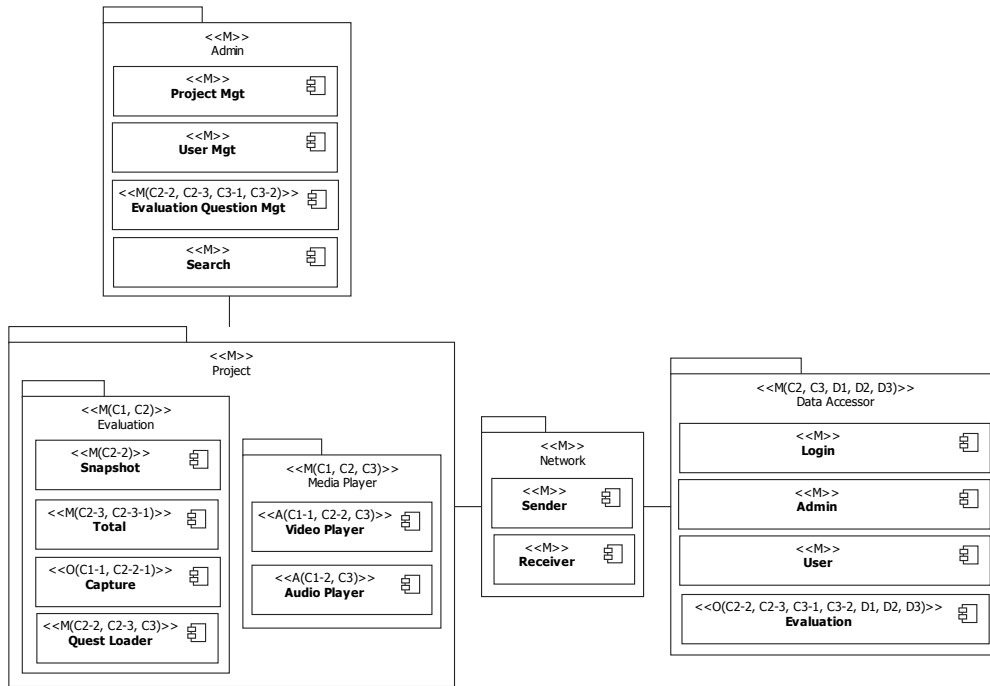


Fig. 5. Architecture Model with Variability of Online Evaluation Classroom Behavior Service product line

Player는 가변성이 필수적인 Media Player의 대안목록으로 존재한다. 그러므로 Video Player와 Audio Player는 구성요소간 가변성규칙이 일치한다.

본 연구에서 제안하는 5가지 규칙을 수업행동분석 제품라인 아키텍처에 적용함으로써 가변성 정보와 추적성 정보의 일관성 및 정확성을 높였으며, 이는 새로운 휘처 조합에 의한 소프트웨어 제품 커스터마이제이션 과정의 정확성과 생산성 품질을 높이는 효과를 보였다.

## 5. 결론

시장적시성(time-to-market) 향상과 비용을 절감을 달성하기 위해서는 소프트웨어 제품라인을 적용하는 것이 방법이 될 수 있다. 하지만 제품라인으로 생산되는 소프트웨어들에 대해서는 좋은 품질을 보장할 수 있어야 한다. 그러므로 휘처모델과 산출물간의 가변성에 대한 일관성을 검증할 수 있는 규칙이 중요하다.

본 연구에서는 휘처를 통해 그 휘처를 구현한 산출물간 가변성에 대한 일관성 검증 규칙 및 산출물들간 가변성에 대한 일관성 검증 규칙을 제시하였었다. 또한, 적용 사례로 수업행동분석 제품라인에 적용하여 휘처와 산출물간 가변성 정보의 일관성을 확보하였으며, 이로써 소프트웨어 제품라인으로부터 제품을 생성하는 과정의 효율성을 높일 수 있다.

## 참고 문헌

- [1] Frank Van Der Linden, Klaus Schmid, Eelco Rommes, Software Product Line in Action, Springer, 2007.
- [2] Tian Hangpei, Gao Deyuan, and Zhu Yian, Gaining Flexibility and Performance of Computing Using Application-Specific Instructions and Reconfigurable Architecture, International Journal of Hybrid Information Technology 2, 2, 2009.
- [3] Felix Bachmann, Paul C. Clements, Variability in Software product lines, Technical report, CMU/SEI-2005-TR-012, Software Engineering Institute, 2005.
- [4] Jean-Christophe TRIGAUX and Patrick HEYMANS, Software Product Lines: State of the art, EPH3310300R0462/ 215315, (2003)
- [5] Mikael Svahnberg, Jilles Van Gorp, Jan Bosch, "A Taxonomy of Variability Realization Techniques," SP&E, 35, 8, (2005)
- [6] IEEE Std 610.12-1990. IEEE Standard Glossary of Software Engineering Terminology. IEEE, New York, 1990.
- [7] S. H. Kim, "Variability Modeling Method for FORM-UML and Consistency Checking Techniques", 2012.
- [8] Kyo C. Kang, Sajoong Kim, Jaejoon Lee, Kijoo Kim, Gerard Jounghyun Kim, Euseob Shin, FORM : A Feature-Oriented Reuse Method with Domain-Specific Reference Architectures, Annals of software engineering. Volume 5. Number 1, 1998.
- [9] NIPA, Applying Guide of Engineering Technology for Software Product Lines based on Software Reuse, 2009.
- [10] Tonny Kurniadi Satyananda, Danhyung Lee, Sungwon Kang,

Formal Verification of Consistency between Feature Model and Software Architecture in Software Product Line, Software Engineering Advances, 2007.

- [11] S. H. Kim, J. A. Kim, "Consistency Checking Rules of Variability between Feature Model and Elements in Software Product Lines", 2012.
- [12] S. H. Kim, J. W. Choi, J. A. Kim, "Construction of Online Classroom Monitoring System", The 34th Fall Conference of KIPS vol. 17, no. 2, pp.421-424, 2010.



### 김 정 아

e-mail : clara@kd.ac.kr

1994 중앙대학교 컴퓨터공학과(공학박사)  
1996~현 재 관동대학교 컴퓨터 교육과 교수  
관심분야: 아키텍처 기반 소프트웨어 설계,  
제품라인 공학 방법론, 프로세스 개선



### 김 세 훈

e-mail : oskywingso@gmail.com

2011 관동대학교 컴퓨터교육과 졸업(학사)  
2012 관동대학교 경영정보학과 졸업(석사)  
2011~현 재 에듀크래프트 대표이사  
관심분야: 소프트웨어 재사용, 소프트웨어 아키텍처